

Методы поиска в пространстве состояний.

Лекция 3.

Специальность : 230105

Поиск на графе.

Все методы перебора в пространстве состояний могут быть смоделированы с помощью следующего теоретико-графового процесса:

Начальная вершина S соответствует описанию начального состояния. Вершины, непосредственно следующие за данной, получаются в результате использования операторов, которые применимы к описанию состояния, ассоциированного с этой вершиной.

Определение 1. Пусть γ - некоторый специальный оператор, который строит все вершины, непосредственно следующие за данной. Будем называть процесс применения оператора γ к вершине раскрытием вершины. Для каждой раскрываемой вершины делается проверка, является ли вершина целевой.

Определение 2. От каждой дочерней вершины к родительской идут указатели, которые позволяют найти путь назад к начальной вершине после обнаружения целевой вершины. Вершины и указатели, построенные в процессе перебора, образуют поддерево всего неявно определенного дерева пространства состояний. Данное поддерево называют *деревом перебора*.

Определение 3. Решающую последовательность образуют операторы, которые связаны с дугами пути от целевой вершины к начальной.

Слепой перебор.

При полном описании процесса перебора необходимо знать порядок раскрытия вершин.

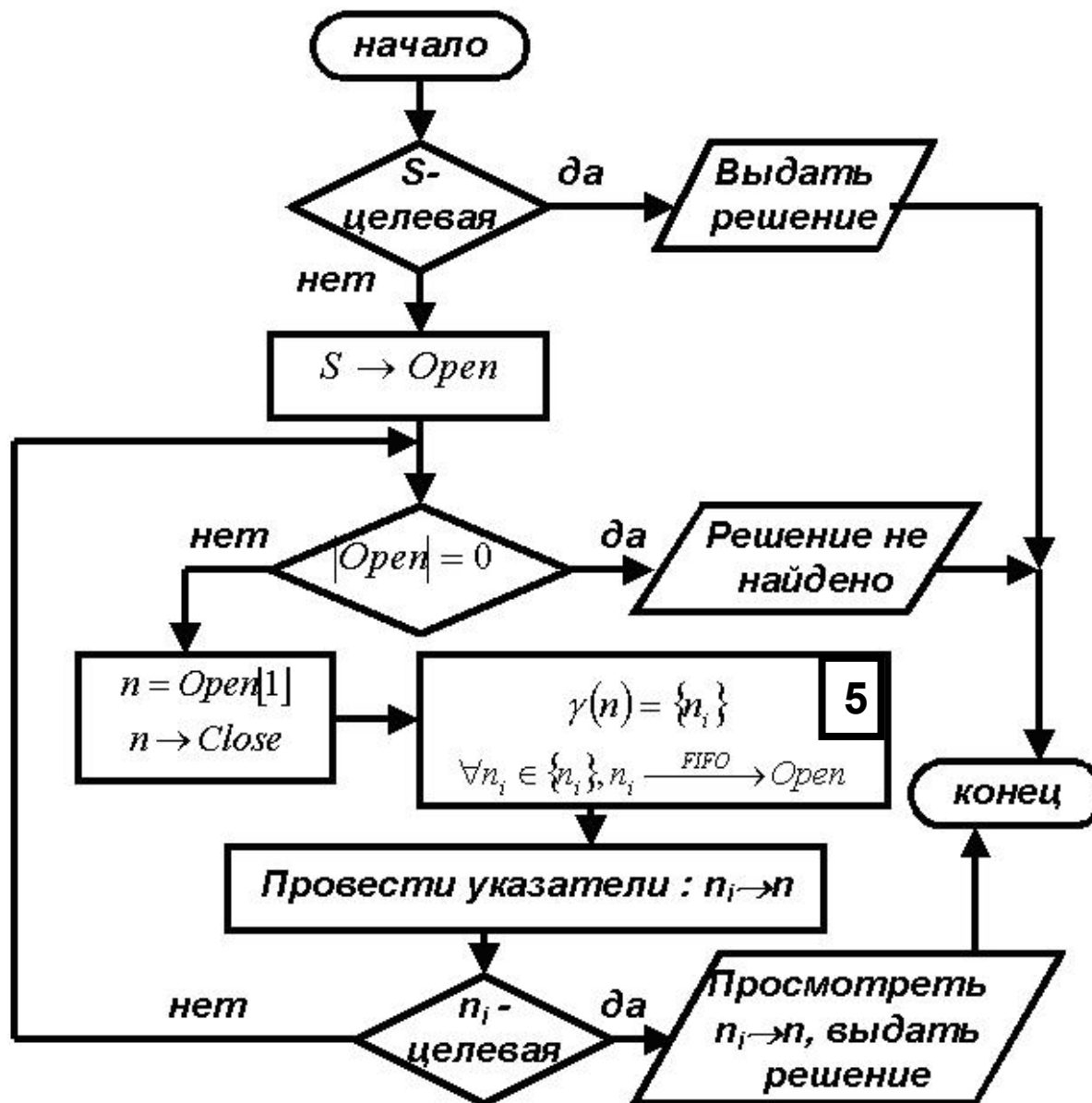
Определение 4. *Полным перебором* называется процесс перебора, при котором раскрытие вершин производится в порядке их порождения.

Определение 5. Процесс перебора в *глубину* (depth-first process) характеризуется тем, что вначале раскрывается та вершина, которая была построена самой последней.

Процессы полного перебора и перебора в глубину называют процедурами слепого перебора, поскольку расположение цели не влияет на порядок раскрытия вершин.

Легко показать, что в методе полного перебора непременно будет найден самый короткий путь к целевой вершине при условии наличия как минимум одного решения. Следует отметить, что при отсутствии решений для случая бесконечного графа реализующий полный перебор алгоритм никогда не завершит свою работу.

Метод полного перебора.



Здесь :

Open – список вершин, к которым еще не применили оператор γ ;

Close – список вершин, к которым уже применялся оператор γ .

Указатели позволяют восстановить путь назад после обнаружения целевой вершины.

На шаге 5 алгоритма вершины помещаются в конец списка *Open*.

Пример : игра в 8.

Пусть задано исходное расположение фишек на поле (см. предыд. лекцию) :

2	8	3
1	6	4
7		5

Целевое расположение :

1	2	3
8		4
7	6	5

Для каждой очередной вершины графа состояний ее дочерние вершины строятся в следующем порядке : первой строится вершина для перемещения пустой клетки влево, второй – для перемещения пустой клетки вверх, третьей – для перемещения пустой клетки вправо, последней – для перемещения пустой клетки вниз. Для каждой новой вершины определяется возможность ее построения. Критерий допустимости построения вершины, дочерней для текущей (допустимости очередного хода) : пустая клетка не должна оказаться за пределами поля, а соответствующее рассматриваемой дочерней вершине расположение фишек не должно быть эквивалентным расположению фишек для вершины, родительской для текущей (возврат к предыдущей конфигурации фишек недопустим). Дерево состояний при использовании полного перебора для игры в 8 представлено в [1] на стр.56. В указанной постановке задачи имеем оптимальное решение из пяти шагов, в процессе поиска которого раскрывается 26 и строится 46 вершин.

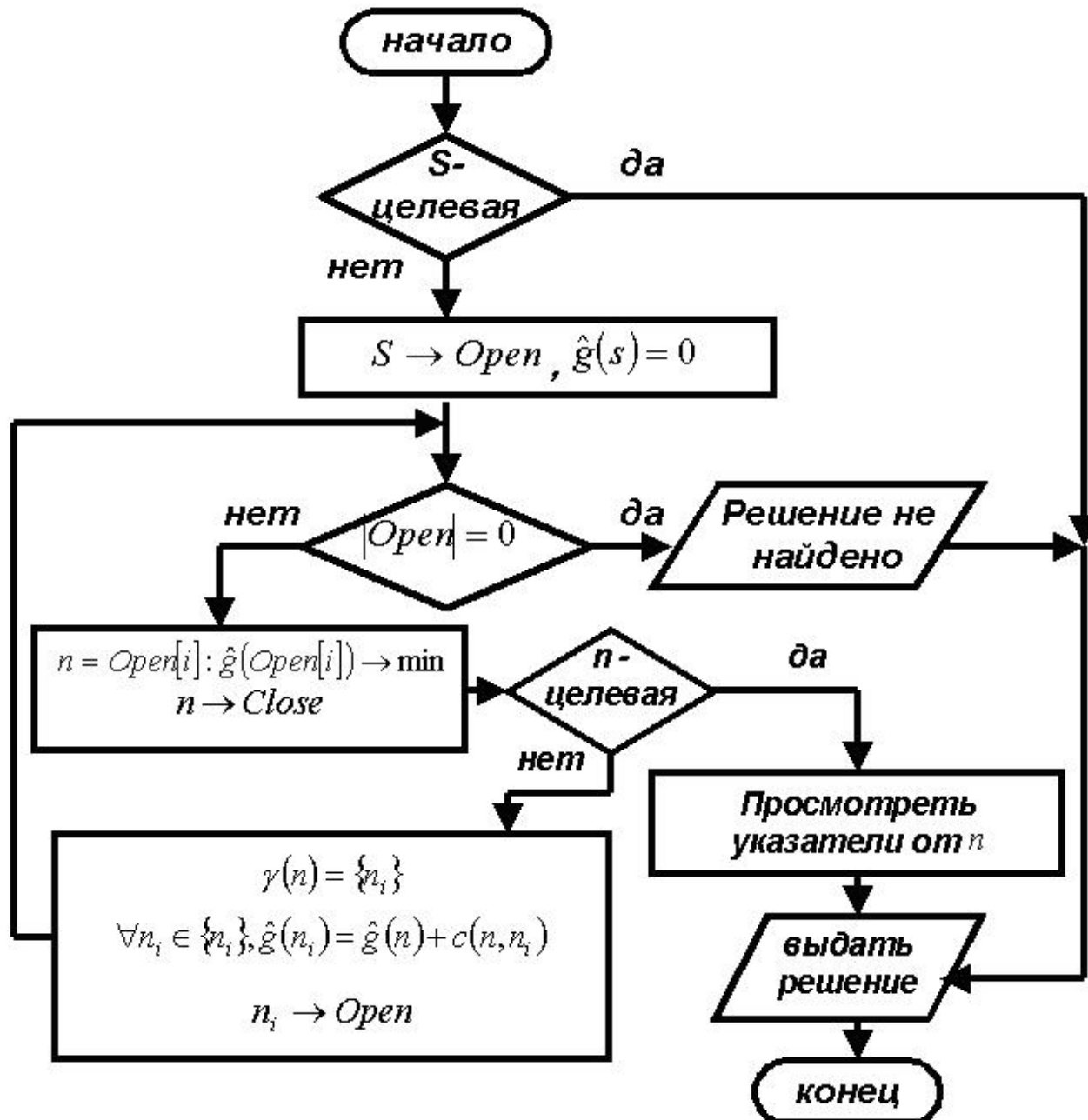
Метод равных цен.

Определение 6. Пусть нам задана функция стоимости $c(n_i, n_j)$ перехода от вершины n_i к некоторой следующей за ней вершине n_j . Метод равных цен характеризуется тем, что для каждой вершины n в дереве перебора необходимо помнить стоимость пути от начальной вершины S к вершине n . Пусть $\hat{g}(n)$ - стоимость пути от вершины S к вершине n в дереве перебора. В методе равных цен вершины раскрываются в порядке возрастания стоимости $\hat{g}(n)$.

Для случая деревьев (не произвольных графов !) перебора в силу единственности пути минимальной длины функция $\hat{g}(n)$ является стоимостью пути наименьшей длины.

Замечание. В то время как в алгоритме поиска по методу полного перебора распространяются линии равной длины пути от начальной вершины, при использовании метода равных цен распространяются линии равной стоимости пути.

Блок-схема алгоритма.



Замечание.

Если положить стоимость каждого ребра в графе состояний равной 1, то поиск пути минимальной стоимости сводится к поиску кратчайшего пути.

Метод перебора в глубину.

Согласно Определению 5, при использовании перебора в глубину в первую очередь раскрываются последние из построенных вершин.

Определение 7. (глубина вершины в дереве).

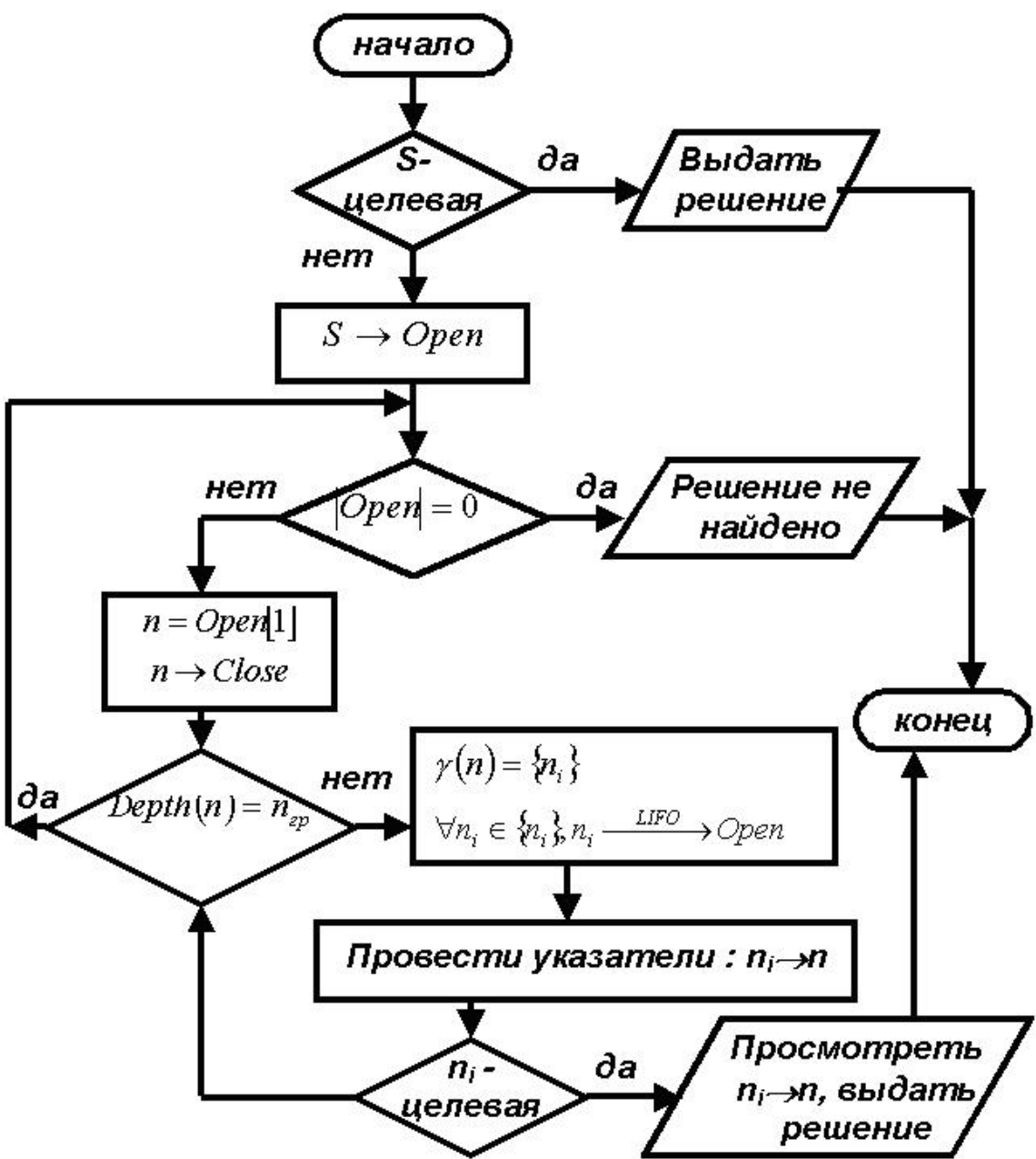
1) Глубина корня дерева равна нулю.

2) Глубина любой последующей вершины дерева равна глубине родительской вершине, увеличенной на 1.

Определение 8. Вершиной с наибольшей глубиной в дереве перебора а некоторый момент служит раскрываемая в данный момент.

Определение 9. *Граничная глубина* при использовании метода перебора в глубину вводится в целях отсечения заведомо бесполезных путей в дереве и организации процедуры *возврата*. При построении вершины с глубиной, превышающей *граничную*, следующей будет раскрыта вершина наибольшей глубины, не превышающей граничного значения.

Блок-схема для метода перебора в глубину.



При поиске целевого расположения фишек игры в 8 при использовании метода перебора в глубину раскрытие отдельной вершины производится аналогично методу полного перебора, построение же дерева состояний ведется в соответствии с приведенным здесь алгоритмом. Граничная глубина выбрана равной 5, это – экспериментально полученное значение длины кратчайшего пути от начальной вершины к целевой при применении метода полного перебора. Дерево состояний при использовании полного перебора для игры в 8 представлено в [1] на стр. 61. В процессе поиска оптимального решения было построено уже 34, а раскрыто 18 вершин.

ПЕРЕБОР НА ПРОИЗВОЛЬНЫХ ГРАФАХ.

При использовании полного перебора для произвольного графа, не являющегося деревом, для каждой вновь построенной вершины следует проверять ее наличие в списках *Open* и *Close*, поскольку она могла появиться там раньше в результате раскрытия некоторой другой вершины. В случае присутствия вновь построенной вершины в *Open* или *Close* ее не нужно вновь помещать в список *Open*.

В целях применимости метода равных цен к произвольному графу сам метод модифицируется следующим образом.

- 1) В случае присутствия вновь построенной вершины n в списке *Open* повторного ее внесения в *Open* не производят, но соответствующая ей величина $\hat{g}(n)$ сравнивается с полученной ранее для другого пути и если значение стоимости для нового пути к n оказывается меньше, то оно запоминается с сопутствующим обновлением указателей.
- 2) В случае присутствия вновь построенной вершины n в списке *Close* сосчитанное ранее значение стоимости пути к ней уже минимально.

Использование эвристической информации.

Определение 10. *Эвристическую* (помогающую найти решение) *информацию* или *эвристику* для задачи, представляемой в виде графа, составляют эмпирические правила, которые позволяют уменьшить объем перебора на основе специфики самой задачи.

Способы использования эвристической информации :

- 1) Модифицированный оператор γ ;
- 2) Размещение вновь построенных вершин в списке Open для метода перебора в глубину;
- 3) Использование оценочных функций для переупорядочивания на каждом шаге списка Open.

В большинстве практических задач представляет интерес *минимизация комбинации* из стоимости пути от начальной вершины к целевой и стоимости перебора для нахождения этого пути. Наибольший интерес представляют методы перебора, при которых указанная комбинация минимизируется, будучи усредненной по всему множеству решаемых задач.

Определение 11. Говорят, что один метод перебора обладает большей эвристической силой, чем другой, если он имеет более низкую усредненную комбинационную стоимость.

Использование оценок функций.

Определение 12. *Оценочная функция* обеспечивает возможность ранжирования вершин – кандидатов на раскрытие с целью выявления той вершины, которая с наибольшей вероятностью находится на оптимальном пути к целевой вершине.

Пусть задана некоторая произвольная функция \hat{f} , которая может быть использована для упорядочения вершин перед их раскрытием.

Обозначим через $\hat{f}(n)$ значение этой функции на вершине n .

Определение 13. Алгоритмом *упорядоченного перебора* (*ordered-search algorithm*) будем называть некоторый переборный алгоритм, в котором для очередного раскрытия выбирается та вершина списка *Open*, для которой значение функции \hat{f} оказывается наименьшим.

Метод упорядоченного перебора подобен методу равных цен, однако для обеспечения его применимости к произвольным графам необходимо предусмотреть перемещения вершины из списка *Close* (с сопутствующим изменением указателей !) в случае нахождения нового пути к вершине со снижением значения величины \hat{f} силу того, что $\hat{f}(n)$ может зависеть от пути из S к n даже для вершин из списка *Close*.

Алгоритм упорядоченного поиска.

Блок-схема - см. приложение.

Следует отметить, что множество порождаемых алгоритмом вершин и указателей образуют дерево (дерево перебора), на листьях этого дерева расположены вершины из списка *Open*.

Рассмотрим работу алгоритма на примере игры в 8. Воспользуемся следующей оценочной функцией :

$$\hat{f}(n) = \hat{g}(n) + W(n), \text{ где} \quad (1)$$

$\hat{g}(n)$ - длина пути в дереве перебора от начальной вершины до вершины n , а $W(n)$ - число фишек, которые лежат не на своем месте в описании состояния, связанного с вершиной n .

Для игры в 8 мы будем предполагать, что с большей вероятностью на оптимальном пути находится та вершина, которая имеет наименьшую оценку.

Построение дочерних вершин для каждой очередной вершины дерева состояний – в соответствии с правилами на слайде 5.

Как видно из представленного на слайде 14 дерева, применение упорядоченного перебора для поиска целевого расположения фишек в игре в 8 дает тот же путь решения, что и метод полного перебора, но применение оценочной функции ведет к значительному сокращению числа раскрытий вершин.

Упорядоченный перебор для игры в 8.

Начальная вершина

2	8	3
1	6	4
7		5

$$n = 0, \hat{g}(0) = 0, W(0) = 4,$$

$$\hat{f}(0) = 0 + 4 = 4$$

2	8	3
1	6	4
	7	5

$$n = 1, \hat{g}(1) = 1, W(1) = 5,$$

$$\hat{f}(1) = 1 + 5 = 6$$

2	8	3
1		4
7	6	5

$$n = 2, \hat{g}(2) = 1, W(2) = 3,$$

$$\hat{f}(2) = 1 + 3 = 4$$

2	8	3
1	6	4
7	5	

$$n = 3, \hat{g}(3) = 1, W(3) = 5,$$

$$\hat{f}(3) = 1 + 5 = 6$$

2	8	3
	1	4
7	6	5

$$n = 4, \hat{g}(4) = 2, W(4) = 3,$$

$$\hat{f}(4) = 2 + 3 = 5$$

2		3
1	8	4
7	6	5

$$n = 5, \hat{g}(5) = 2, W(5) = 3,$$

$$\hat{f}(5) = 2 + 3 = 5$$

2	8	3
1	4	
7	6	5

$$n = 6, \hat{g}(6) = 2,$$

$$W(6) = 4,$$

$$\hat{f}(6) = 2 + 4 = 6$$

	8	3
2	1	4
7	6	5

$$n = 7,$$

$$\hat{g}(7) = 3,$$

$$W(7) = 3,$$

$$\hat{f}(7) = 3 + 3 = 6$$

2	8	3
7	1	4
	6	5

$$n = 8,$$

$$\hat{g}(8) = 3,$$

$$W(8) = 4,$$

$$\hat{f}(8) = 3 + 4 = 7$$

	2	3
1	8	4
7	6	5

$$n = 9,$$

$$\hat{g}(9) = 3,$$

$$W(9) = 2,$$

$$\hat{f}(9) = 3 + 2 = 5$$

2	3	
1	8	4
7	6	5

$$n = 10,$$

$$\hat{g}(10) = 3,$$

$$W(10) = 4,$$

$$\hat{f}(10) = 3 + 4 = 7$$

1	2	3
	8	4
7	6	5

$$n = 11, \hat{g}(11) = 4, W(11) = 1,$$

$$\hat{f}(11) = 4 + 1 = 5$$

**Целевая
вершина**

1	2	3
8		4
7	6	5

$$n = 12,$$

$$\hat{g}(12) = 5,$$

$$W(12) = 0,$$

$$\hat{f}(12) = 5 + 0 = 5$$

1	2	3
7	8	4
	6	5

$$n = 13,$$

$$\hat{g}(13) = 5,$$

$$W(13) = 2,$$

$$\hat{f}(13) = 5 + 2 = 7$$

Свойства оценочной функции.

Определим оценочную функцию \hat{f} так, чтобы значение $\hat{f}(n)$ для любой вершины n представляло собой сумму оценки величины пути минимальной стоимости от начальной вершины s к вершине n и оценки величины пути минимальной стоимости от вершины n к целевой вершине. Таким образом, $\hat{f}(n)$ представляет собой оценку величины пути минимальной стоимости при условии, что этот путь проходит через вершину n .

Пусть функция $k(n_i, n_j)$ дает действительную величину пути минимальной стоимости между двумя произвольными вершинами n_i и n_j

Величина пути минимальной стоимости от вершины n_i до цели :

$$h(n_i) = \min_{n_j \in T} k(n_i, n_j), \text{ где } T - \text{множество целевых вершин} \quad (2)$$

Стоимость оптимального пути от начальной вершины s до цели при условии, что он проходит через вершину n :

$$f(n) = g(n) + h(n), \text{ где} \quad (3)$$

$g(n)$ - действительная стоимость оптимального пути от начальной вершины s до некоторой произвольной вершины n . Отсюда следует, что действительная стоимость оптимального пути от вершины s к цели (без дополнительных ограничений на промежуточные вершины) $f(s) = h(s)$
 $\hat{g}(n) \geq g(n)$ - по определению.

Оптимальный алгоритм перебора.

Будем рассматривать введенную нами ранее оценочную функцию \hat{f} в качестве оценки функции f . Пусть наша оценка дается выражением :

$$\hat{f}(n) = \hat{g}(n) + \hat{h}(n), \text{ где} \quad (4)$$

\hat{g} — оценка для g , а \hat{h} — оценка для h . В содержательной интерпретации $\hat{g}(n)$ есть стоимость пути от начальной вершины S до n в дереве перебора, получается после суммирования стоимостей дуг на пути по указателям от n назад к S .

При вычислении оценки $\hat{h}(n)$ для $h(n)$ используется любая эвристическая информация, связанная с самой задачей. В дальнейшем будем называть \hat{h} эвристической функцией.

Назовем алгоритм упорядоченного перебора, в котором используется оценочная функция $\hat{f}(n) = \hat{g}(n) + \hat{h}(n)$, алгоритмом A^* .

При $\hat{h} \equiv 0$ алгоритм A^* совпадает с алгоритмом равных цен.

Теперь покажем, что если \hat{h} — произвольная нижняя граница для h , то алгоритм A^* также находит оптимальный путь к цели.

Определение 14. Будем называть алгоритм перебора допустимым, если для произвольного графа он оканчивает свою работу построением оптимального пути к целевой вершине (при условии существования такого пути).

Допустимость алгоритма A^* .

Лемма 1. Если для всех вершин n графа перебора выполняется условие $\hat{h}(n) \leq h(n)$, то в любой момент времени до того, как алгоритм A^* закончит свою работу, на любом оптимальном пути P от вершины S к цели существует открытая вершина n' , для которой $\hat{f}(n') \leq f(s)$

Замечание. Открытой вершиной мы будем называть любую вершину списка *Open*, вершины из списка *Close* мы будем называть закрытыми, а незакрытой – такую вершину, которая либо находится в списке *Open*, либо еще не была построена в процессе перебора.

Доказательство леммы 1 следует из определений оценочной функции \hat{f} и стоимости $f(n')$ оптимального пути от начальной вершины S до цели при условии, что он проходит через вершину n'

Теорема 1. Если для всех вершин n графа перебора выполняется условие $\hat{h}(n) \leq h(n)$ и если стоимости всех дуг превосходят некоторое малое положительное число δ , то алгоритм A^* допустим.

Доказательство теоремы 1 строится методом от противного с применением доказанной леммы 1 для трех случаев, когда алгоритм A^* не находит оптимальный путь от начальной вершины к целевой : алгоритм завершил работу и не нашел целевой вершины, нет окончания работы алгоритма и окончание на целевой вершине без достижения минимальной стоимости.

Оптимальность алгоритма A^* .

Определение 15. Будем говорить, что алгоритм A более информирован, чем алгоритм B , если используемая в алгоритме A эвристическая информация применительно ко всем вершинам n , не являющихся целевыми, позволяет вычислять такую нижнюю границу для $h(n)$, которая всюду строго больше, чем та, которая вычисляется по эвристической информации, используемой в алгоритме B .

Будем предполагать, что для любых двух вершин m и n , для которых существует действительное значение пути минимальной стоимости $k(m, n)$, выполняется условие : $\hat{h}(m) - \hat{h}(n) \leq k(m, n)$. Иными словами, разность между оценками стоимости путей от любой пары вершин m и n до цели должна быть нижней границей стоимости оптимального пути от m до n .

Мы сделаем *предположение о непротиворечивости* применения эвристической информации при вычислении \hat{h} во всех вершинах графа.

Используя предположение о непротиворечивости, мы можем доказать в общем случае, что когда при работе алгоритма A^* происходит раскрытие некоторой вершины, то оказывается, что оптимальный путь к этой вершине уже найден. Таким образом, возможность переоткрытия вершин, предусмотренная шагами (*) и (**) алгоритма упорядоченного поиска, оказывается излишней и может быть удалена из него, если удовлетворяется предположение о непротиворечивости.

Выбор эвристической функции.

Использование условия $\hat{h} \equiv 0$ гарантирует допустимость, но ведет к слепому перебору. Выбор в качестве \hat{h} наибольшей из возможных нижних границ для h приводит к тому, что раскрывается наименьшее число вершин, при котором еще сохраняется допустимость.

Для ряда задач эвристическая сила алгоритма может быть повышена ценой отказа от допустимости при использовании в качестве \hat{h} некоторой функции, не являющейся нижней границей для h .

Пример : для игры в 8 функция $\hat{h}(n) = W(n)$ есть нижняя граница для $h(n)$. Но гораздо лучшую оценку дает функция $\hat{h}(n) = P(n)$, где $P(n)$ - сумма расстояний каждой фишки от “своего места” (без учета фишек, расположенных на ее пути). Недостаток : не учитывается должным образом трудность обмена местами двух соседних фишек. Наиболее удачна для игры в 8 следующая оценка :

$$\hat{h}(n) = P(n) + 3 * S(n), \text{ где} \quad (5)$$

$S(n)$ – число очков, учитывающее порядок расположения фишек. Для его вычисления применительно к заданной конфигурации необходимо последовательно просмотреть все нецентральные фишки и за каждую фишку, за которой не идет та фишка, которая должна идти в целевой конфигурации, начисляется 2 очка, в противном случае – 0 очков. За фишку в центре – 1 очко. Однако в этом случае функция \hat{h} не дает нижней границы для h .

Оценочная функция для игры в 8.

Рассмотрим пример использования предложенной нами эвристической функции (5) в оценочной функции (4) для следующего случая игры в 8 :

*Начальная
конфигурация*

2	1	6
4		8
7	5	3



*Целевая
конфигурация*

1	2	3
8		4
7	6	5

Построение дочерних вершин для каждой очередной вершины дерева состояний – в соответствии с правилами на слайде 5. Рассмотрим вторую и третью из раскрываемых вершин ([1], стр.79).

2:

2	1	6
	4	8
7	5	3

$$n = 2, P(n) = 1 + 1 + 3 + 2 + 1 + 2 + 1 + 0 = 11$$

$$S(n) = 2 + 2 + 2 + 2 + 2 + 2 + 2 + 1 = 15$$

$$\text{Согласно (5), } \hat{h}(n) = 11 + 3 * 15 = 11 + 45 = 56, \hat{g}(n) = 1, \hat{f}(n) = 57$$

3:

2	1	6
4	8	
7	5	3

$$n = 3, P(n) = 1 + 1 + 3 + 1 + 2 + 2 + 1 + 0 = 11$$

$$S(n) = 2 + 2 + 2 + 2 + 2 + 2 + 2 + 1 = 15$$

$$\text{Согласно (5), } \hat{h}(n) = 11 + 3 * 15 = 11 + 45 = 56, \hat{g}(n) = 1, \hat{f}(n) = 57$$

Использование эвристической функции (5) применительно к рассматриваемому случаю игры в 8 дает решающий путь длиной 18 шагов.

Эвристическая сила алгоритма упорядоченного поиска.

Зависит от трех важных факторов :

- 1) Стоимость пути;
 - 2) Число раскрываемых в процессе поиска пути вершин;
 - 3) Объем требуемых для подсчета значений функции \hat{h} вычислений.
- Выбор соответствующей \hat{h} позволяет получать компромисс между этими 3-мя факторами, максимизировать эвристическую силу алгоритма.

Абсолютный минимум числа раскрытий вершин обеспечивает такая функция \hat{h} , которая в точности совпадает с h .

В случае функции \hat{h} , отличной от нижней границы для h эвристическая сила алгоритма может быть увеличена как благодаря уменьшению общего числа раскрываемых вершин (ценою отказа от допустимости), так и благодаря уменьшению объема вычислений.

В ряде случаев эвристическая сила функции \hat{h} может быть повышена умножением ее на некоторую положительную константу, большую 1 (при достаточно большом множителе $\hat{g}(n) \equiv 0$). Такой выбор приведет к недопустимому (согласно Теореме 1), но, тем не менее, способному удовлетворительно работать алгоритму (когда просто требуется просто найти какой-нибудь путь к цели, а его стоимость значения не имеет).

Критерии качества работы методов перебора.

1) Целенаправленность перебора – позволяет узнать, в какой мере перебор идет в направлении цели, а не ведется по нежелательным направлениям. Определяется соотношением :

$$P = \frac{L}{T} \quad , \text{ где} \quad (6)$$

L – длина найденного пути до цели, а T – общее число построенных в течение перебора вершин (включая целевую, но исключая начальную).

Повышение информированности оператора γ дает более высокую целенаправленность, которая в пределе стремится к единице.

Слепой перебор характеризуется малыми значениями P .

Как правило, увеличение L ведет к еще более быстрому росту T .

2) Фактор эффективного ветвления B – основан на представлении о дереве с глубиной, равной длине пути и общим числом вершин T (6), причем у каждой вершины имеется в точности B дочерних вершин.

B связан с L и T соотношениями :

$$B + B^2 + \dots + B^L = T \quad (7)$$

$$\frac{B}{B-1} * (B^L - 1) = T \quad (8)$$

Взаимосвязь критериев качества.

Зависимость B от T для различных значений L при использовании фактора эффективности ветвления может быть представлена линейной диаграммой ([1], стр. 87). Величина B , близкая к 1, соответствует точно направленному к цели перебору, который мало ответвляется на другие направления. Кустообразный граф перебора характеризуется высоким значением B .

Целенаправленность P связана с B и длиной пути L соотношением :

$$P = \frac{L * (B - 1)}{B * (B^L - 1)} \quad (9)$$

В той мере, в какой B мало зависит от длины пути до цели, величина B может быть использована для предсказания того, сколько вершин было бы построено при переборе той или иной длины.

Литература.

Нильсон Н. Искусственный интеллект : Пер. с англ. - М.: Мир, 1973. С. 52-90.